Lecture 2

CSE 331

Feb 3, 2021

Enroll on Piazza

https://piazza.com/buffalo/spring2021/cse331

Read the syllabus and hw policy CAREFULLY!

Syllabus Quiz

Options	
View handin history	
View writeup	
Download handout	

🕐 Due: February 28th 2021, 3:47 pm

📅 Last day to handin: February 28th 2021, 5:47 pm

No graded material will be handed back till you pass the syllabus quiz!
195 (out of 271) already completed!

Academic Integrity

Question 1: Sharing my answers to this syllabus quiz with other 331 students

○ Is OK if I do it to help out a friend

○ It does not matter since there is no grade attached with it

○ Is an academic integrity violation and should not be done

○ Is an academic integrity violation but I can take the chance

Question 2: Penalty for academic violation in CSE 331 is an automatic

○ Warning and a chance to make-up

○ A zero in the assignment AND a letter grade reduction (for first violation across all CSE courses) and an F in the course (for 2nd violation across all CSE courses)

○ A zero in the corresponding assignment and nothing else

○ Expulsion from UB

You can submit the following now

Assignments

Homework 0	Quizzes
Q1 part (a) [Number of perfect matchings]	Syllabus Quiz
Q1 part (b)	
Q3 (Structured Matrix Vector Multiplication)	

You all should be on Autolab now
(let me know if not!)

HWs

[50 pts] Q1 (easier) is a proof-based question; two parts

[25 pts] Q2 (harder) is a proof-based question; two parts

[25 pts] Q3 is a programming question; autograded

HWs due by 8:00pm on Fridays
(assigned a week before)

NO LATE SUBMISSIONS!!!

Support page

CSE 331 Care Package

Starting Fall 2019, CSE 331 will be assuming more background material was covered in CSE 250 (and CSE 191). In particular, starting Spring 2020, CSE 331 will assume a non-trivial coverage of proofs and other related material in CSE 191 and CSE 250. While we make this transition, this page collect materials that were covered in previous incarnations of CSE 331 but will no longer be covered going forward (this also includes material that are not covered in CSE 191/250). We hope that this page is helpful if you took the older version of CSE 191/250 or you took the equivalent courses in another school.

The Topics

Below we collect the topics that we will no longer cover in CSE 331 (but were covered as late as Fall 17 or Fall 18):

- Asymptotic Notation
- Trees
- Topological Ordering
- Minimizing Maximum Lateness

C++ vs Java/Python

Use Java/Python if you are just as comfortable with as C++

Use a VM with g++ installed for Ubuntu

We recommend that you install a VM that runs `g++` on Ubuntu. In particular, we recommend that you use Jaric Zola's ☑ VM system that he created for his CSE 250 course ☑.
If you have questions on Jaric's setup, please do **NOT** contact him: email `cse-331-staff@buffalo.edu` instead.

If you still prefer using your own system, we would still recommend that you test your code in the VM system above before submitting to Autolab.

Allowed Sources

Allowed sources

You can **ONLY** use the following sources for reference once you start working on the homework problems:

1. **the Kleinberg-Tardos textbook,**

> ### Other textbooks are not allowed
>
> While you can use other textbooks (e.g. those listed in the syllabus) to better understand the lecture material, you **cannot** use them once you start working on the homeworks.

2. **any material linked from this webpage or the CSE 331 piazza page (including any discussion in the Q&A section),**

> ### One-click rule
>
> When using webpages that are allowed as sources, you **cannot** click on link on that source. (Otherwise within a constant number of clicks one can reach any webpage one wants.)

3. **specific *mathematical* result from a previous course,**
4. **anything discussed in the lectures, recitations and/or office hours** and
5. **any notes that you might have taken during class or recitation.**

Everything else is not allowed

Note that the above list covers all the allowed sources and **everything else is not allowed**. In particular, *YOU ARE NOT SUPPOSED TO SEARCH FOR SOLUTIONS ON THE*

… even for programming Q

- All discussions and posts on piazza ⌀ .

Basic programming references

C++ Sources

- cppreference.com ⌇ (and all pages within the website).

Java Sources

- Oracle Java Documentation ⌇ (and all pages within the website).

Python Sources

- Python 3.5.2 documentation ⌇ (and all pages within the website).

Asymptotic Analysis

- Big-O cheat sheet ⌇ .

Wikipedia Pages

Below are some approved Wikipedia pages (in addition to those that are already linked to in other pages in the CSE 331 Fall 2018 web page.

- Gale Shapley algorithm ⌇ .
- DFS ⌇ .
- Dijkstra's Algorithm ⌇ .
- Prim's algorithm ⌇

Why do proofs?

Makes you think logically about problems and solutions

From an engineer who works on Google Maps:

Proving that the algorithm I am implementing is correct helps me identify corner cases

Why should we do proofs?

We will focus a lot on proofs in CSE 331. In this document I will motivate why doing proofs is good even though you might not do proofs for a living. ▪ While doing this, we will also go through examples of how to write algorithm ideas and details as well as proof ideas and details (which you will need to write in your homework solutions).

Some reasons to do proofs

In this section, I will lay out some reasons why I think it is beneficial for you guys to do proofs. The first two are probably more along the lines of "if you do proofs for a living" situation. The rest of the reasons should be valid for all of you. I will try and make the reasons as concrete as possible: in the next section, we will consider algorithms for the specific problem of generating all permutations (recall that we previously had punted on designing an algorithm for this problem).

Sometimes you might not have a choice

One of the easiest way to verify an algorithm idea you have is to code up the algorithm and then test it on some (say random) inputs. However, sometimes this might not be a choice. E.g. if you work on Quantum Computing ⌐, then you do not have a quantum computer to run your quantum code on! So currently pretty much the only choice you have is to *prove* that your algorithm is indeed correct. For example, one of the crowning achievements of quantum computing is Shor's algorithm ⌐ to computes the factors of large numbers efficiently on a quantum computer (that recall does not exist yet!). (You might also want to read Scott Aaronson's ⌐ high level description of Shor's algorithm ⌐.) The reason why factoring large numbers ⌐ is important is that if one can solve this problem efficiently then one can break the RSA cryptosystem ⌐. RSA is used everywhere (e.g. when you use your credit card online, RSA is used to make the transaction secure), so this is a big deal.

https://cse.buffalo.edu/~erdem/cse331/support/proofs/index.html

A more subtle incorrect "proof"

Brad Pitt has a beard

waleg.com

Every goat has a beard
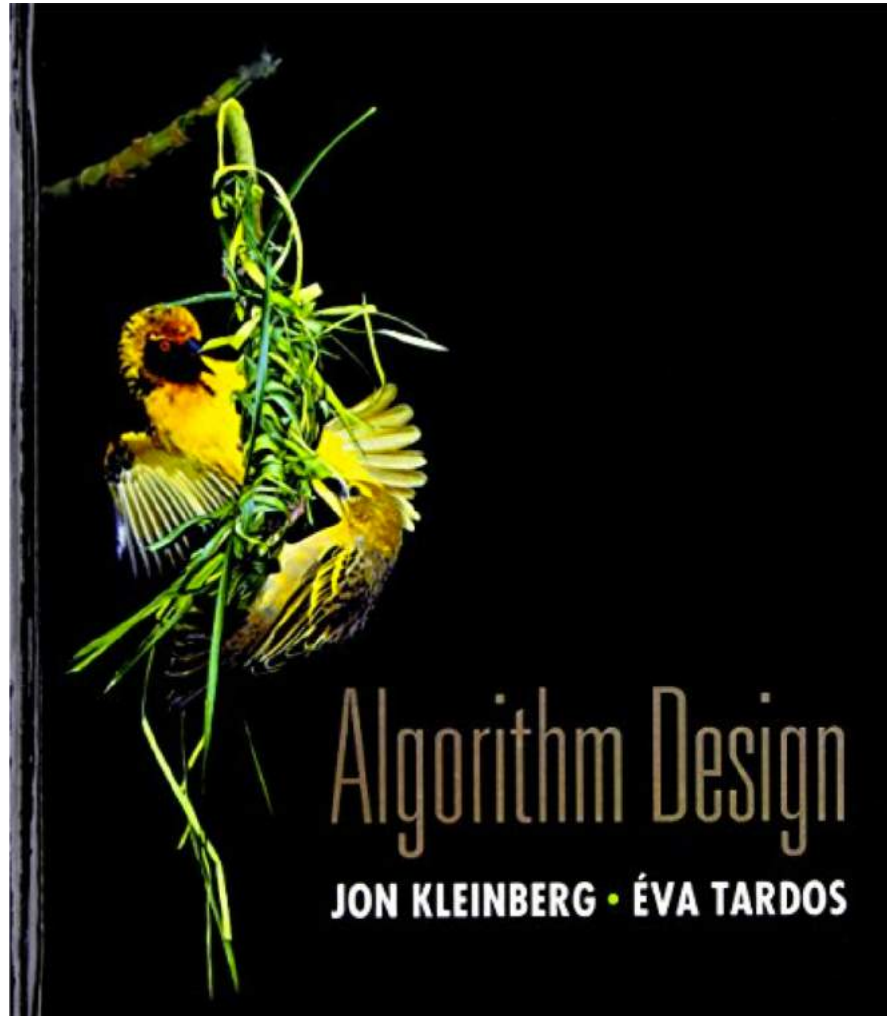

animaldiversity.org

Hence, Brad Pitt is a goat.

More chances to recover

Lowest two HW scores will be dropped

If you do better on the final exam than the mid-term exam then only final exam score will count

Follow the Textbook

CSE 331 Support Page

This page contains certain webpages that students taking CSE 331 might find useful.

The material is roughly divided into two parts: one on (primarily mathematical) background material and one of common mistakes that students generally make.

Disclaimer

Please note that this material is intended as a support material. It is not meant as a replacement for actually having taken background courses like CSE 116, 191 or 250 nor is this meant to be exhaustive. I'll try my best to make these as comprehensive as possible but that might take some time.

Background material

CSE 331 will need a fair bit of math: most of which you must have seen earlier. However, if you have not used those material for a bit then you might be a bit rusty. The pages linked below are some notes that I wrote up that might help you refresh the material that you might have seen in CSE 116, 191 or 250. Also some of the pages are meant to motivate why we do some things a certain way in CSE 331. Finally sometimes (but not often!) we will use material that might not have been covered in previous courses and we did not have much time to cover in class: these pages will fill in those gaps.

Common Mistakes

Here we collect some common mistakes that students make in CSE 331 material (and sometimes more than once). The hope is to list these common pitfalls so that you can avoid them!
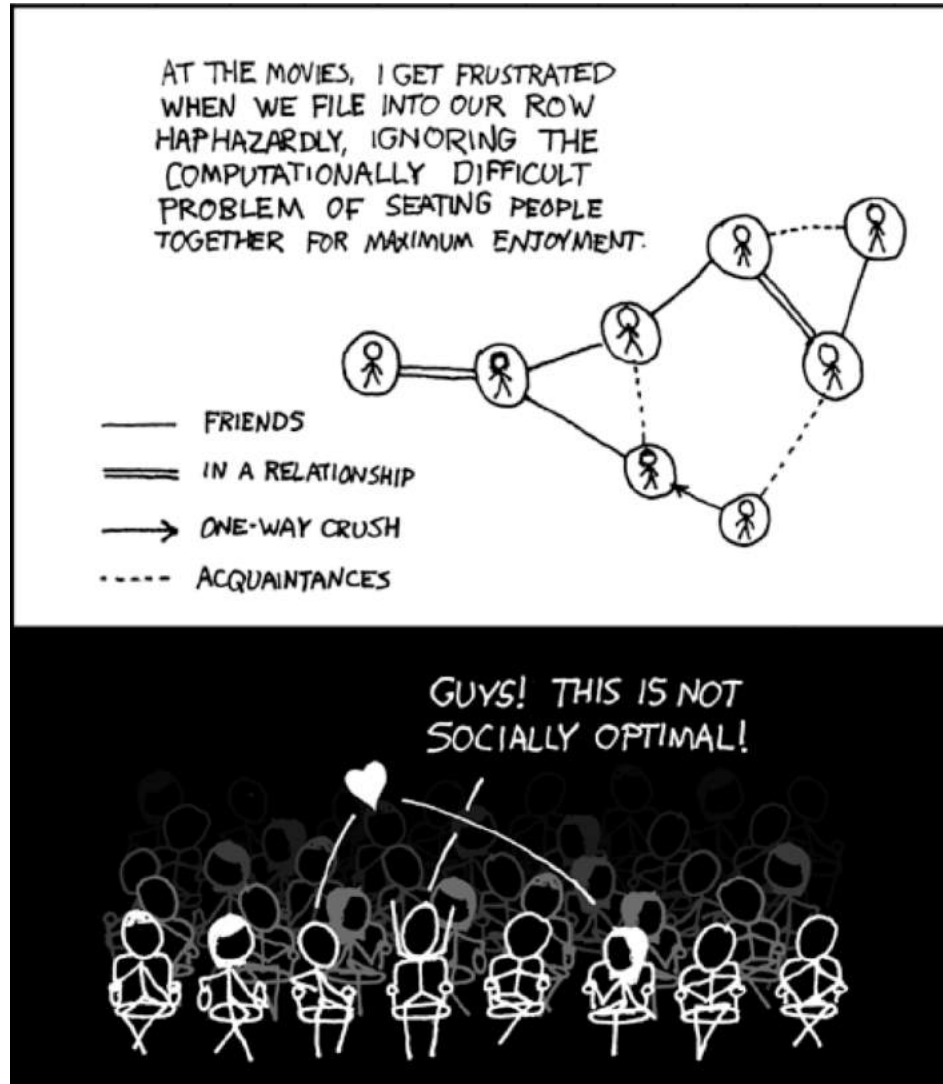
Other Resources

Below we collect other 331 related material that do not neatly fall into the two left category:

- Visualizing Algorithms.
- Algorithms via Examples.

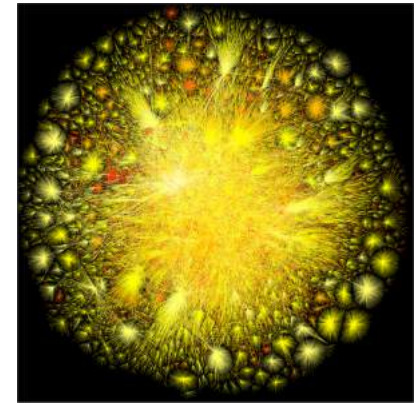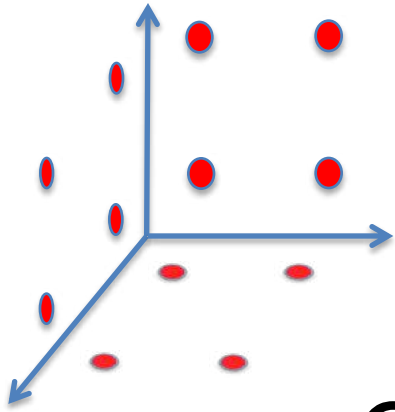https://cse.buffalo.edu/~erdem/cse331/support/index.html

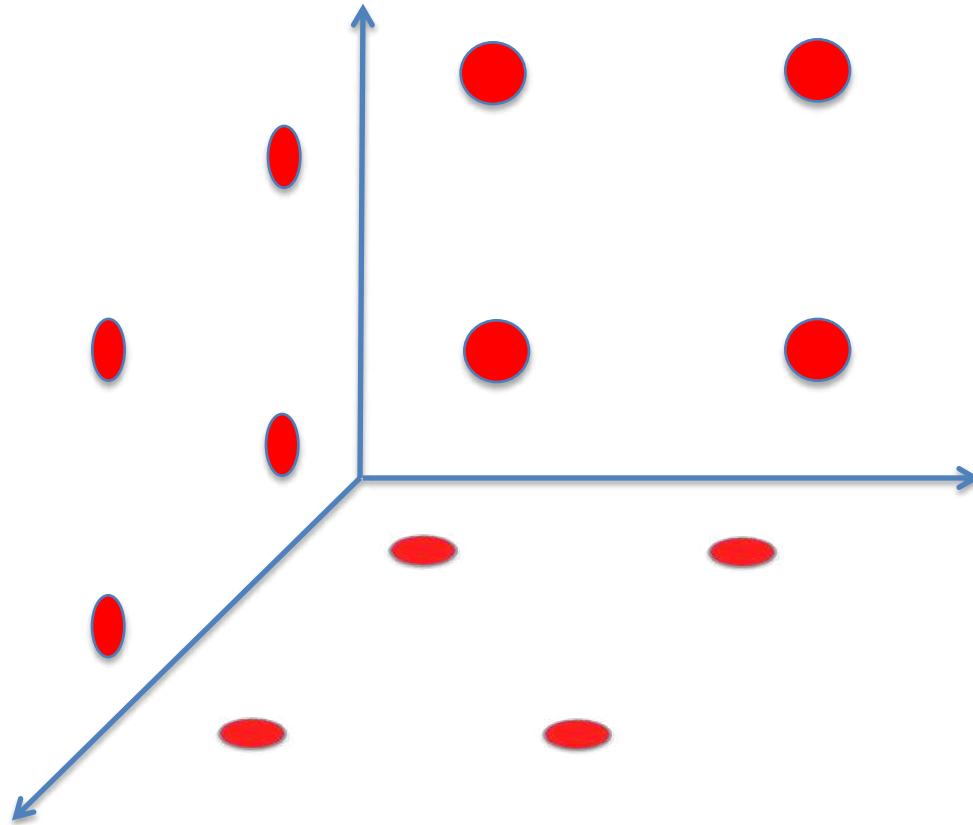# Questions/Comments?

This course: how to solve problems!

Why should I care ?

Combining Shadows to Understanding the network
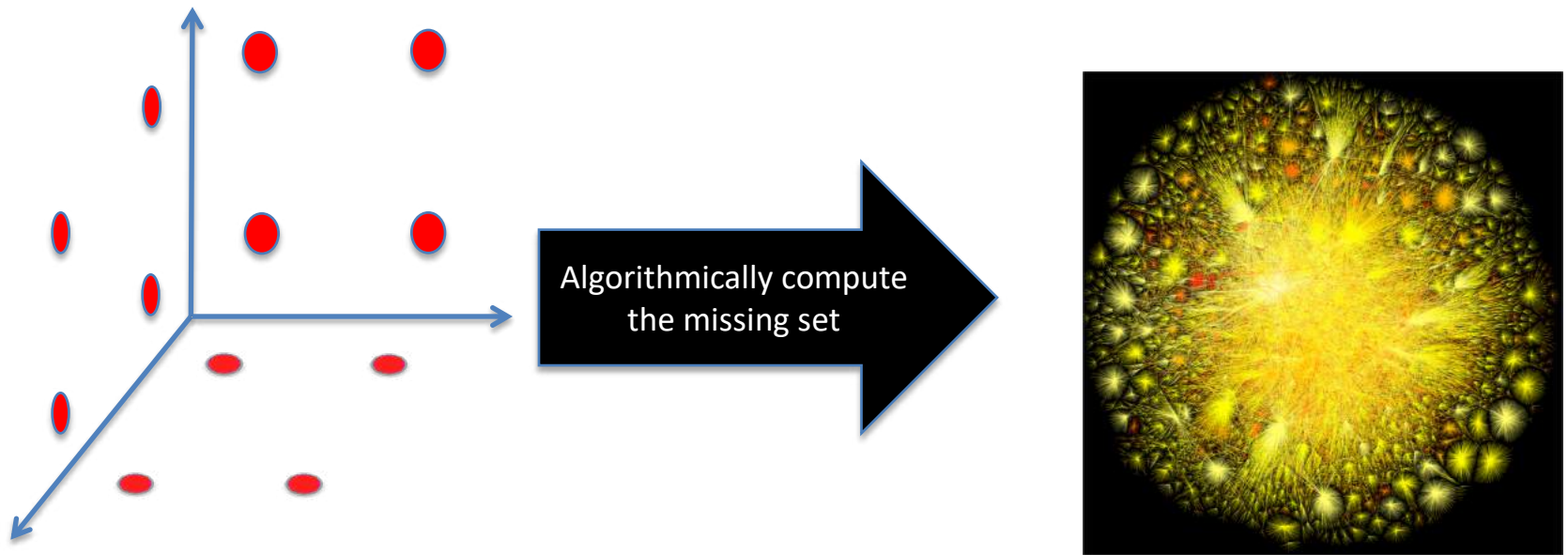
The key technical problem

Given the three projections, what is the largest size of the original set of points?

Conquering Shadows to Conquering the Internet

Algorithmically compute the missing set

The proof is in the performance

10x faster

Better algorithm with little hacking will beat a worse algorithm with tons of hacking

The key technical problem
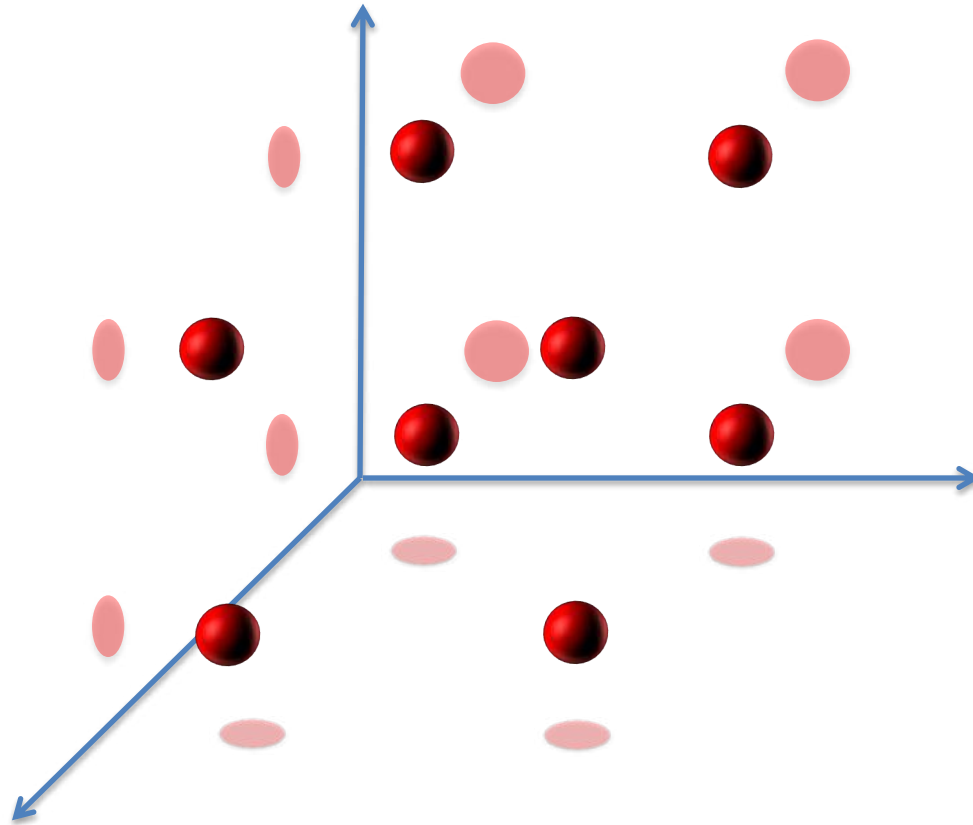
Highly trivial: $4^3 = 64$ Still trivial: $4^2 = 16$ Correct answer: $4^{1.5} = 8$

From someone who got a Google job

"You can let your algorithms class know that the phone interviews are essentially like **a difficult algorithms test**.

Lots of data structures, specifying the algorithm, analyzing the run time and space requirements... And all on the phone and **you're supposed to talk through your thought process**."

Coding jobs will be done by AI

Coding jobs will be done by AI

MIT News
ON CAMPUS AND AROUND THE WORLD

Browse or Search

FULL SCREEN

Researchers have developed a flexible way of combining deep learning and symbolic reasoning to teach computers to write short computer programs. Here, Armando Solar-Lezama (left), a professor at CSAIL, speaks with graduate student Maxwell Nye.

Photo: Kim Martineau

Toward artificial intelligence that learns to write code
Researchers combine deep learning and symbolic reasoning for a more flexible way

So am I doomed?

There will still be room for high level *algorithmic* thinking!

European Association for
Theoretical Computer Science

HOME   ABOUT   SEARCH   CURRENT   ARCHIVES

Home > No 125: June 2018 > **Lamport**

If You're Not Writing a Program, Don't Use a Programming Language

Leslie Lamport, *Distributed Computing & Education Column by Juraj Hromkovic, Stefan Schmid*

Today, programming is generally equated with coding. It's hard to convince students who want to write code that they should learn to think mathematically, above the code level, about what they're doing. Perhaps the following observation will give them pause. **It's quite likely that during their lifetime, machine learning will completely change the nature of programming. The programming languages they are now using will seem as quaint as Cobol, and the coding skills they are learning will be of little use. But mathematics will remain the queen of science, and the ability to think mathematically will always be useful.**

Why care about algorithms?

Driving directions

Why care about algorithms?

Computing Bestsellers on the fly

Why care about algorithms?

Booking cheapest air tickets

Why care about algorithms?

Google searches

Why care about algorithms?
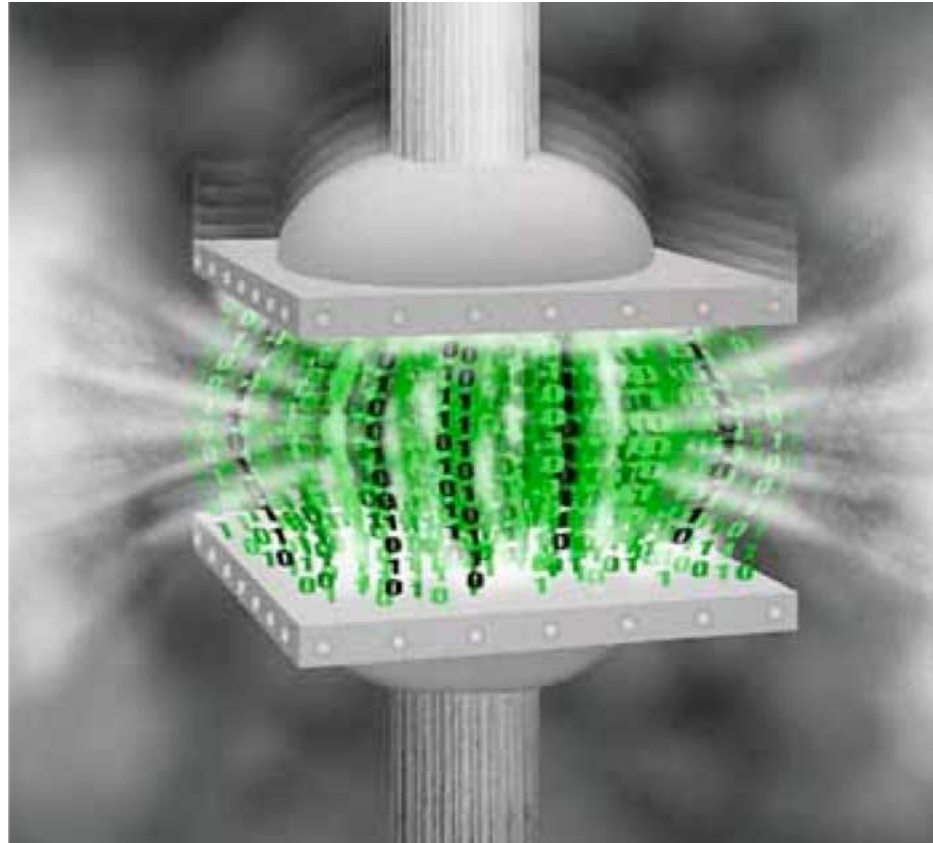
Data compression

(And I could) go on…

Find out for yourself

Video project: Video on ethical impacts of algorithm. Groups of size = 3

CSE 331 Video Project

Spring 2021

Details and motivations for the video project.

Motivation

CSE 331 is primarily concerned with the technical aspects of algorithms: how to design them and then how to analyze their correctness and runtime. However, algorithms are pervasive in our world and is common place in many aspects of society. The main aim of the video project is to have you explore in some depth social implications of algorithms.

Just to give two examples for such implications:

- Algorithms are pervasive in financial transactions and these algorithms have consequences beyond just trading:

Questions/Comments?